DL\_LAB1\_Report

310512073 電控所 曾致崴

1. Introduction

在這次的LAB1中主要是針對Backpropagation以及網路如何更新權重，並了解其中基本原理如何更接近目標，在一開始先對整體網路訓練的流程了解後，先架設Feedforward network，並利用得到的output利用Backpropagation計算出各項參數的gradient值最後update 權重，完成一次的訓練，而在訓練的參數上對於結果來說也是很重要的，例如learning rate的調控會影響整體模型在訓練時震盪的程度，以及好不好收斂，hidden layer 的unit要設多少，也會影響網路的收斂速度，若是太大或太小也會對網路造成不同的影響。

整題程式執行方式

執行程式環境 : Windows 11

Source code 底下總共有三個py檔 分別為 main.py、nets.py、utiles.py，其中main才是主要執行的程式，執行程式會出現acc 95%的兩個case

1. Experiment setups
   1. Sigmoid functions

Text

Description automatically generated

**圖 1**

* 1. Neural network

在本次的lab中我主要是使用class 的方式去建置整體的model，而其中有一些初始化的參數，如下圖2所示

1. lr (learning rate) : 0.001
2. epoch : 1000
3. random\_seed : 0 此參數是為了確保每一次training的結果相同，在init\_weight中有定義
4. hid\_layer\_size : 是指hidden layer中要用多少個node

Text

Description automatically generated

**圖 2**

在下圖3中定義了6個參數 的大小

: 2 x hidden layer node

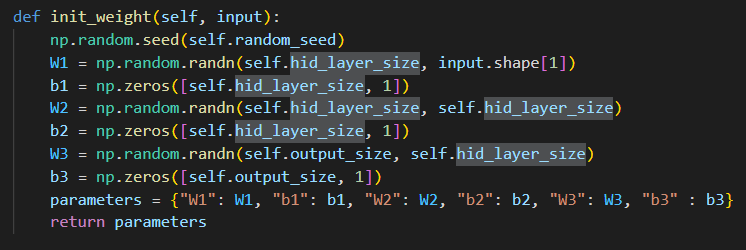
: hidden layer node x hidden layer node

: hidden layer node x number of classes

: 1 x hidden layer node

: 1 x hidden layer node

: 1 x number of classes



**圖 3**

在Network 的設計上這次是以為架構主要是有3個linear層以及3個activation function ReLU、Sigmoid、Sigmoid，如下圖4所示，圖中左邊為網路架構流程圖，右邊為網路架構示意圖

Diagram

Description automatically generated

**圖 4 網路架構示意圖**

Text

Description automatically generated

**圖 5 feedforward network 程式圖**

Text

Description automatically generated

**圖 6 forward network 用到的function**

* 1. Backpropagation

Backpropagation 主要可以分為兩個部分，分別是propagation、weight update

1. Propagation

因為 以及 ，利用chain rule 反著推，從最後一層是sigmoid開始，對偏微分如下式

接著對偏微分(linear 層)

接著對偏微分(Sigmoid 層)

接著對偏微分(linear 層)

接著對偏微分(ReLU 層)=>對小於0的直接給值=0其餘一樣

依照這個寫成程式如下圖7所示

Text

Description automatically generated

**圖 7 Backpropagation 程式圖**

1. Weight update

在weight update就是，原本的值 ”減” backpropagation 所得到的gradient各項參數值乘上learning rate 對weight 做更新，如下圖8所示

Text

Description automatically generated

**圖 8 weight update 程式圖**

1. Results of your testing
   1. Screenshot and comparison figure

Task 1 predict linear label ( lr = 0.01, epoch=500, random\_seed=0, hidden unit = 10)

Chart, scatter chart

Description automatically generated

圖 9

Task 2 predict XOR label (lr = 0.01, epoch=300, random\_seed=0, hidden unit =100)

Chart, scatter chart

Description automatically generated

圖 10

* 1. Accuracy of prediction

Task 1 predict linear label

Graphical user interface, text

Description automatically generated

圖 11

Task 2 predict XOR label

Graphical user interface, text

Description automatically generated

圖 12

* 1. Learning curve

Task 1 predict linear label

Graphical user interface, chart

Description automatically generated

Task 2 predict XOR label

Chart

Description automatically generated with medium confidence

* 1. Others

Text

Description automatically generated圖 13 train function

在程式的底下有加上一個限制條件當loss小於0.001時就會跳掉training，因為當loss 小於一定值時去計算backpropagation時會使分母太小造成結果變成NaN的情況，所以為了避免這樣的狀況發生，在此有加上保護機制

1. Discussion
   1. Different learning rate

目前固定hidden layer unit = 10，epoch = 500，learning rate 以10倍做調整

Task 1 predict linear label

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| hidden unit | epochs | Learning rate | Acc | End epoch |
| 10 | 500 | 0.0001 | 59% | 500 |
| 10 | 500 | 0.001 | 91% | 500 |
| 10 | 500 | 0.01 | 99% | 300 |
| 10 | 500 | 0.1 | 99% | 25 |

在這個task 底下可以看到learning rate 變大整體收斂的速度也變快，但超過包護機制所設定threshold，就會開始發散

Task 2 predict XOR label

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| hidden unit | epochs | Learning rate | Acc | End epoch |
| 10 | 500 | 0.0001 | 47.62% | 500 |
| 10 | 500 | 0.001 | 80.96% | 500 |
| 10 | 500 | 0.01 | 85.71% | 500 |
| 10 | 500 | 0.1 | 57.14% | 500 |

在這個task 底下，可以明顯的看到learning rate 越大，有先變好的趨勢但最後是整個發散掉的情況，在train loss上可以明顯的看到loss到大概100 epochs時有收斂，但後續就發散掉

Chart, histogram

Description automatically generated

圖 14 loss 沒收斂圖

* 1. Different numbers of hidden units

Task 1 predict linear label

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| hidden unit | epochs | Learning rate | Acc | End epoch |
| 1 | 500 | 0.01 | 63% | 500 |
| 10 | 500 | 0.01 | 100% | 330 |
| 50 | 500 | 0.01 | 99% | 37 |
| 100 | 500 | 0.01 | 98% | 57 |

在task 1上的表現，與前一個章節調整lr得到相同的結論，當hidden unit越多，網路會越快收斂，但同時當今天超過保護機制的狀況會造成發散nan的情況

Task 2 predict XOR label

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| hidden unit | epochs | Learning rate | Acc | End epoch |
| 1 | 500 | 0.01 | 47% | 500 |
| 10 | 500 | 0.01 | 85.71% | 500 |
| 50 | 500 | 0.01 | 90.48% | 500 |
| 100 | 500 | 0.01 | 100% | 260 |

在hidden layer unit = 1 的情況下，整體的模型因為不構complex所以在一開始就發散掉，如下圖所示，當hidden unit 開始變多後模型才開始會收斂的比較好

A picture containing graphical user interface

Description automatically generated

圖 15 hidden unit = 1 loss 圖

Chart

Description automatically generated

圖 16 hidden unit = 10 loss 圖

Chart

Description automatically generated

圖 17 hidden unit = 50 loss 圖

* 1. Without activation functions

1. Extra
   1. Implement different optimizers

在cost function上我使用的是cross entropy

Text, logo

Description automatically generated

* 1. Implement different activation functions

ReLU

